Digital Chart of the World (DCW) To The Joint Tactical Simulation (JTS) Translator Technical Report

Tobi Sellekaerts

May 1998

Prepared for:

7th Army Training Command (7th ATC)
Grafenwoehr, Germany

Terrain Modeling Project Office (TMPO)
Washington, D.C.

Under Contract:
DAJA 22-96-D0069

Prepared by:

Logicon, RDA
Grafenwoehr, Germany

Introduction

The 95th Military Police Battalion recently requested a large playbox of JTS terrain of low resolution. Due to these unique requirements, TerraSim decided to create the terrain from Digital Chart of the World (DCW). As a DCW importer already existed for generating CBS terrain, it was simply modified to create JTS compatible Arc/Info coverages.

Using DCW for JTS terrain construction is much easier than creating terrain from scratch; the program is rather comprehensive. The user simply enters four corner coordinates or chooses 5 degree squares from a map of the world. No advance preparation other than determining the four corners and obtaining the DCW disks is necessary. The program's main drawbacks, however, are that the corner coordinates may only be entered in decimal degrees of latitude and longitude, not UTM meters, and that DCW contains no vegetation information outside of North America. If vegetation is required by the customer, it must be created using traditional terrain building methods.

Program Flow

This AML uses no arguments; it prompts the user for all necessary information. First, the user is asked to assign this terrain a name of five letters or less, which will be used to create a workspace into which the completed JTS coverages will be copied when the program is finished. (More information on the workspace naming convention below.) Next, the program asks the user how he would like to choose a geographical area: either choosing 5 degree cells from a map or entering coordinates at the keyboard. If the user chooses the map option, and map of the entire world appears and the user simply selects the boxes he would like to use. (The source information for the map and grid which appear on the screen is contained in two coverages called world_map and world_tile which reside on Italy and are accessible if this program is run from any TerraSim workstation that has Italy's drives mounted. Please contact TerraSim to obtain these two coverages.)

If the user chooses the keyboard entry option, he types in four coordinates to determine the playbox extent (upper left longitude, upper left latitude, lower right longitude, lower right latitude). The program determines the extent of the final playbox and creates the standard JTS terrain clip coverage. With this clip coverage, the program zcreate.aml (which has been embedded at this point) creates empty JTS coverages (i.e. `paved, hydro`) which are ready to accept data.

The program determines which DCW cells (and CD-ROMs) will be needed for data extraction and prompts the user to insert the appropriate CD-ROM when necessary. Once the appropriate DCW coverages have been extracted, the cells of each coverage type are combined and borders dissolved. Features are then extracted from the DCW coverages and placed into the JTS coverages based on the following table:

DCW name & abbreviation		Arc/Info feature type	DCW attribute item	DCW attribute value	Extracted JTS coverage
drainage	`DN`	line	`dnlntype`	`1,4`	`hydro`
drainage	`DN`	polygon	`dnpytype`	`1,2`	`lakes`
land cover	`LC`	polygon	`lcpytype`	`1`	`swamp`
land cover	`LC`	polygon	`lcpytype`	`3`	`vineyard`
land cover	`LC`	polygon	`lcpytype`	`6`	`lakes`
political / oceans	`PO`	polygon	`popytype`	`2`	`lakes`
populated places	`PP`	polygon	`pppytype`	`1`	`urban`
railroads	`RR`	line	`rrlntype`	`1,2,8`	`rail`
roads	`RD`	line	`rdlntype`	`1,8`	`bahn`
roads	`RD`	line	`rdlntype`	`2,8`	`paved`
roads	`RD`	line	`rdlntype`	`3,8`	`loose`

The JTS coverages are placed in a new workspace beneath the workspace in which the AML is run. The workspace is called

```
jts<name>#
```

The letters `jts` do not change. `<name>` is the name given by the user for this terrain build. `#` is a number which starts at `0` and goes up incrementally each time the program is run. If you ran `dcwjts.aml` and gave `test` as your name, the program would create a workspace for the final JTS files called

```
      jtstest0
```

If you then ran the program from the same workspace and again specified `test` as the name, the program would create a workspace called

```
      jtstest1
```

and not overwrite the first workspace created. This is somewhat redundant naming as the user could specify a different name each time the program is run, but it was a very helpful feature during debugging.

These coverages can then be used to create the JTS terrain file as if they had been generated by any other source data.

The source code of the program follows the Reference section.

Reference

For more information about the JTS coverage requirements and terrain file format, please see the *Joint Tactical Simulation (JTS) Terrain Building Model (JTBM)* written by Dave Baumgartner and Erik Patton available from TerraSim.

For information about Digital Chart of the World attributes and attribute values, see *VPFView Users Manual for the DCW* available from the National Imagery and Mapping Agency (NIMA).

The source data used for generating JTS terrain in this program is Digital Chart of the World (DCW) Edition 1 July 1993 available from NIMA. (If the areal coverage extent and directory naming convention does not change, future editions of DCW should also be compatible.)

For more information on the DCW to CBS program on which this translator is based, see the *CBS 1.5.4 Translator Technical Report* written by Tobi Sellekaerts available from TerraSim.

```
/* dcwjts.aml

/* ======================================================
/* AML to convert DCW data to JTS Terrain.
/* Modified from dcwarc.aml in April 1998.
/* History:
/* Written in June and July 1995 for CUBIC TSI by Tobi
Steinberg.
/* Modified in April and May 1996 by Tobi Steinberg.
/* Rewritten (argh...) in September 1996 by Tobi.
/* The structure of the program is as follows:  First, the
user chooses the
/* areas he would like to include in his playbox.  This can
be done
/* in two ways:
/* 1. Select some boxes from a 5 by 5 degree grid laid over
a map of
/*         the world.
/*    2. Select the corners of the playbox by lat/long
coordinates.
/* Second, the needed coverages are extracted from the DCW
CD-ROM.  Third,
/* these coverages are merged and interior borders
dissolved.
/*
/* I have been rather selective about which coverages are
imported from DCW.
/*
/* Major revisions started 9 April 1998 by Tobi Sellekaerts.
I plan to:
/*      - remove UTM coordinate entry option
/*      - debug so the program won't bail
/*      - in general, clean up the ugly thing
/*      - world_tile and world_map no longer need to be in
the same workspace
/*      - checking to be sure it works with the new DCW
disks
/* ======================================================

&echo &on
&messages &on
&s junk [close -all]

&if [exists dcwarc[date -tag].wat -file] &then
   &s delwat [delete dcwarc[date -tag].wat]
&watch dcwarc[date -tag].wat; &ty [date -VFULL]

&s cdpath /cdrom/cdrom0
```

```
&if [exists /data2/cbs/world_tile -cover] &then
  &s source /data2/cbs/
&else &s source /net/italy/data2/cbs/

display 9999 3

&do file &list xtilelist.txt xformat.txt             /*
Deleteing needed files
   &if [exists %file% -file] &then &s junk [delete %file% -
file]
&end

&label badlength
&ty Please make sure that the name of your area is less than
&ty five characters long.
&ty
&ty
&s name [response 'What would you like to call this area?'
area]
&if [length %name%] gt 5 &then &goto badlength

&type When the program is done importing and combining the
data,
&s view [response 'do you want to view the coverages? [y] or
n' y]

/* Deleting coverages and files needed by the program.

&do cov &list xtiles xtilesutm xclipcov xcliproj xtemp
      &if [exists %cov% -cover] &then kill %cov% all
&end

&do cov &list ae dn hy hs of pp ds rr rd po lc
   &do type &list pt ln py
      &if [exists x%cov%%type% -cover] &then
        kill x%cov%%type% all
      &if [exists %name%%cov%%type% -cover] &then
        kill %name%%cov%%type% all
   &end
&end

&do file &list xpy.txt xpt.txt xln.txt
   &if [exists %file% -file] &then &sys rm %file%
&end

/* *******************************************************
/* Deciding what areas of the world are to be imported into
ARC
```

```
/* *******************************************************
/* Variables, files, coverages used:
/*    xtilelist.txt - list of DCW tile names to be used
/*    total_t(n) - the total number of tiles chosen from
each CD
/*    xtiles - a coverage containing the selected tiles
/*    method - the way the user would like to choose the
area

&ty You may choose your geographical area in one of two
ways:
&ty      1. View a map of the world with 5 by 5 degree boxes
placed
&ty         across its surface and choose a group of boxes
covering
&ty         your area of interest.
&ty      2. Specity the upper left and lower right corners
of your
&ty         area of interest by latitude and longitude
&ty         coordinates.

&label method
&s method [response 'How would you like to select your area?
[1] or [2]' 1]
&if %method% ne 1 and %method% ne 2 &then &do
   &ty Please enter a 1 or a 2.
   &goto method
&end

&if %method% = 1 &then &do
ae
   &label choose
   ec %source%world_tile
   me def
   ef poly
   de poly
   bc %source%world_map 5
   be arc
   draw
      &ty Choose squares to select the area that you would
like to
      &ty include in your JTS terrain.
      &ty Please choose a contiguous area and press 9 when
finished.
   &label select
   sel many
   &if [show number selected] = 0 &then &do
      &ty Please choose at least one tile.
      &goto select
   &end
   put xtiles
   save

   ec xtiles
   build
   mape xtiles
   draw
   &s decision [response 'Is this the area you wish to use?
[y]' y]
   &if %decision% = y &then &do
      ef poly
         sel cd = 1
         &s total_t1 [show number selected]
         sel cd = 2
         &s total_t2 [show number selected]
         sel cd = 3
         &s total_t3 [show number selected]
         sel cd = 4
         &s total_t4 [show number selected]
   &end
   &else &do
      ec %source%world_tile
      kill xtiles
      y
      &goto choose
   &end
   save
quit

tables
   sel xtiles.pat
   sort cd
   unload xtilelist.txt tile_name columnar xformat.txt
   sort xtiles-id
quit

&sys rm xformat.txt

&end

/* *********************************************************
/* Selecting an upper left and lower right corner.
/* *********************************************************
/* Coverages, files, and variables used:
/*    cs - type of coordinate system used to input corner
locations
```

```
/*    zone - utm zone the coordinates fall into
/*    ulx - upper left x coordinate
/*    uly - upper left y coordinate
/*    lrx - lower right x coordinate
/*    lry - lower right y coordinate
/*    xclipcov - coverage with the playbox extents
/*    xcliproj - playbox extents coverage projected
/*    xtemp - temporary coverage used to project another
coverage
/*    hemi - the hemisphere the playbox resides in
/*    lat, long, lat(n), long(n) - used to determine the
geographic
/*           center of the playbox area chosen

&if %method% = 2 &then &do
    &s hemi 0
    &ty Please enter coordinates from a WGS84 spheroid and
datum.
    &ty Please enter all values in decimal degrees.
Latitudes in the
    &ty southern hemisphere and longitudes in the eastern
hemisphere
    &ty should be entered as negative values.

    &ty You will be entering coordinates for the upper left
and
    &ty lower right corners of the box only.

    &label badulx
    &s ulx [response '     The x value of the upper left
corner']
    &if %ulx% gt 180 or %ulx% lt -180 &then &do
        &ty Please enter a number between -180 and 180.
        &goto badulx
      &end

    &label baduly
    &s uly [response '     The y value of the upper left
corner']
    &if %uly% gt 90 or %uly% lt -90 &then &do
        &ty Please enter a number between -90 and 90.
        &goto baduly
      &end

    &label badlrx
    &s lrx [response '     The x value of the lower right
corner']
    &if %lrx% gt 180 or %lrx% lt -180 &then &do
```

```
        &ty Please enter a number between -180 and 180.
        &goto badlrx
      &end

    &label badlry
    &s lry [response '     The y value of the lower right
corner']
    &if %lry% gt 90 or %lry% lt -90 &then &do
        &ty Please enter a number between -90 and 90.
        &goto badlry
      &end

    create xclipcov                 /* Creating a clip
coverage using the
    tables                          /* coordinates entered
above.
    sel xclipcov.tic
    add
    1
    %ulx%
    %uly%
    2
    %lrx%
    %uly%
    3
    %lrx%
    %lry%
    4
    %ulx%
    %lry%
    ~
    quit

    display 0
    ae
    ec xclipcov
    ef arc
    coordinates keyboard
    add
    2 %ulx% %uly%
    1 %lrx% %uly%
    1 %lrx% %lry%
    1 %ulx% %lry%
    2 %ulx% %uly%
    ~
    coordinates mouse
    save
    quit
    build xclipcov poly

    projectdefine cover xclipcov
    projection geographic
    units dd
    spheroid wgs84
    parameters
    copy xclipcov xcliproj
  &end

build xcliproj poly
clip %source%world_tile xcliproj xtiles
build xtiles

display 0

ae
ec xtiles
ef poly
sel cd = 1
&s total_t1 [show number selected]      /* number of cells
from CD #1... etc.
sel cd = 2
&s total_t2 [show number selected]
sel cd = 3
&s total_t3 [show number selected]
sel cd = 4
&s total_t4 [show number selected]
quit

tables
sel xtiles.pat
sort cd
resel area gt 0
unload xtilelist.txt tile_name columnar xformat.txt
sel xtiles.pat
sort xtiles-id
quit

&sys rm xformat.txt

/* ***********************************************************
/* Extracting the chosen areas and necessary coverages from
CD-ROMs
/* ***********************************************************
/* Variables, files, and coverages used:
/*     tile - tile number/names as read out of the tile list
file
/*      tile2 - tile name after backslashed corrected
/*      cover - tile name without any slashes

&ty Extracting the chosen areas and necessary coverages from
CD-ROM

&s tilelist [open xtilelist.txt os -read]

   &if %total_t1% ne 0 &then &do
      &if ^ [exists %cdpath%/dcw/noamer -directory] &then
&do
         &sys eject
         &ty Please insert DCW Disk #1 into the CD-ROM drive
         &pause
      &end
      &do loop1 = 1 &to %total_t1%
         &s tile [read %tilelist% rs]
         &s a [locase [substr %tile% 1 1]]
         &s b [locase [substr %tile% 3 1]]
         &s c [locase [substr %tile% 5 2]]
         &s tile2 %a%/%b%/%c%
         &s cover %a%%b%%c%
         &do foo &list dn hy hs of pp ds lc  /* coverages
with point and other features
            &if [exists %cdpath%/dcw/noamer/%foo%/%tile2% -
directory] AND ~
              ^ [exists x%foo%%cover% -cover] AND ~
              ^ [exists x%foo%%cover%pnt -cover] &then &do
                  vpfimport cover
%cdpath%/dcw/noamer/%foo%/%tile2% x%foo%%cover%
            &end
            &if [exists %cdpath%/dcw/noamer/%foo%/%tile2% -
directory] AND ~
              ^ [exists x%foo%%cover% -cover] AND ~
              [exists x%foo%%cover%pnt -cover] &then &do
                  kill x%foo%%cover%pnt all
                  vpfimport cover
%cdpath%/dcw/noamer/%foo%/%tile2% x%foo%%cover%
            &end
            &if [exists %cdpath%/dcw/noamer/%foo%/%tile2% -
directory] AND ~
              [exists x%foo%%cover% -cover] AND ~
              ^ [exists x%foo%%cover%pnt -cover] &then &do
                  kill x%foo%%cover% all
                  vpfimport cover
%cdpath%/dcw/noamer/%foo%/%tile2% x%foo%%cover%
            &end
         &end
         &do foo &list rr rd po           /* coverages with
other features
           &if [exists %cdpath%/dcw/noamer/%foo%/%tile2% -
directory] AND ~
             ^ [exists x%foo%%cover% -cover] &then &do
                 vpfimport cover
%cdpath%/dcw/noamer/%foo%/%tile2% x%foo%%cover%
               &end
         &end
      &end
     &end


   &if %total_t2% ne 0 &then &do
      &if ^ [exists %cdpath%/dcw/eurnasia -directory] &then
&do
         &sys eject
         &ty Please insert DCW Disk #2 into the CD-ROM drive
         &pause
      &end
      &do loop2 = 1 &to %total_t2%
         &s tile [read %tilelist% rs]
         &s a [locase [substr %tile% 1 1]]
         &s b [locase [substr %tile% 3 1]]
         &s c [locase [substr %tile% 5 2]]
         &s tile2 %a%/%b%/%c%
         &s cover %a%%b%%c%
         &do foo &list dn hy hs of pp ds lc  /* coverages
with point and other features
            &if [exists %cdpath%/dcw/eurnasia/%foo%/%tile2%
-directory] AND ~
              ^ [exists x%foo%%cover% -cover] AND ~
              [exists x%foo%%cover%pnt -cover] &then &do
                  kill x%foo%%cover%pnt all
                  vpfimport cover
%cdpath%/dcw/eurnasia/%foo%/%tile2% x%foo%%cover%
                &end
            &if [exists %cdpath%/dcw/eurnasia/%foo%/%tile2%
-directory] AND ~
              [exists x%foo%%cover% -cover] AND ~
              ^ [exists x%foo%%cover%pnt -cover] &then &do
                  kill x%foo%%cover% all
                  vpfimport cover
%cdpath%/dcw/eurnasia/%foo%/%tile2% x%foo%%cover%
                &end
            &if [exists %cdpath%/dcw/eurnasia/%foo%/%tile2%
-directory] AND ~
```

```
              ^ [exists x%foo%%cover% -cover] AND ~
              ^ [exists x%foo%%cover%pnt -cover] &then &do
                  vpfimport cover
%cdpath%/dcw/eurnasia/%foo%/%tile2% x%foo%%cover%
                &end
         &end

         &do foo &list rr rd po           /* coverages with
other features
           &if [exists %cdpath%/dcw/eurnasia/%foo%/%tile2% -
directory] AND ~
             ^ [exists x%foo%%cover% -cover] &then &do
                 vpfimport cover
%cdpath%/dcw/eurnasia/%foo%/%tile2% x%foo%%cover%
               &end
         &end
      &end
     &end

   &if %total_t3% ne 0 &then &do
      &if ^ [exists %cdpath%/dcw/soamafr -directory] &then
&do
         &sys eject
         &ty Please insert DCW Disk #3 into the CD-ROM drive
         &pause
      &end
      &do loop3 = 1 &to %total_t3%
         &s tile [read %tilelist% rs]
         &s a [locase [substr %tile% 1 1]]
         &s b [locase [substr %tile% 3 1]]
         &s c [locase [substr %tile% 5 2]]
         &s tile2 %a%/%b%/%c%
         &s cover %a%%b%%c%
         &do foo &list dn hy hs of pp ds lc  /* coverages
with point and other features
            &if [exists %cdpath%/dcw/soamafr/%foo%/%tile2% -
directory] AND ~
              ^ [exists x%foo%%cover% -cover] AND ~
              ^ [exists x%foo%%cover%pnt -cover] &then &do
                  vpfimport cover
%cdpath%/dcw/soamafr/%foo%/%tile2% x%foo%%cover%
            &end
            &if [exists %cdpath%/dcw/soamafr/%foo%/%tile2% -
directory] AND ~
              ^ [exists x%foo%%cover% -cover] AND ~
              [exists x%foo%%cover%pnt -cover] &then &do
                  kill x%foo%%cover%pnt all
                  vpfimport cover
%cdpath%/dcw/soamafr/%foo%/%tile2% x%foo%%cover%
            &end
            &if [exists %cdpath%/dcw/soamafr/%foo%/%tile2% -
directory] AND ~
              [exists x%foo%%cover% -cover] AND ~
              ^ [exists x%foo%%cover%pnt -cover] &then &do
                  kill x%foo%%cover% all
                  vpfimport cover
%cdpath%/dcw/soamafr/%foo%/%tile2% x%foo%%cover%
            &end
         &end

         &do foo &list rr rd po           /* coverages with
other features
           &if [exists %cdpath%/dcw/soamafr/%foo%/%tile2% -
directory] AND ~
             ^ [exists x%foo%%cover% -cover] &then &do
                 vpfimport cover
%cdpath%/dcw/soamafr/%foo%/%tile2% x%foo%%cover%
               &end
         &end
      &end
   &end

   &if %total_t4% ne 0 &then &do
      &if ^ [exists %cdpath%/dcw/sasaus -directory] &then
&do
      &sys eject
         &ty Please insert DCW Disk #4 into the CD-ROM drive
         &pause
      &end
      &do loop4 = 1 &to %total_t4%
         &s tile [read %tilelist% rs]
         &s a [locase [substr %tile% 1 1]]
         &s b [locase [substr %tile% 3 1]]
         &s c [locase [substr %tile% 5 2]]
         &s tile2 %a%/%b%/%c%
         &s cover %a%%b%%c%
         &do foo &list dn hy hs of pp ds lc  /* coverages
with point and other features
            &if [exists %cdpath%/dcw/sasaus/%foo%/%tile2% -
directory] AND ~
              ^ [exists x%foo%%cover% -cover] AND ~
              ^ [exists x%foo%%cover%pnt -cover] &then &do
                  vpfimport cover
%cdpath%/dcw/sasaus/%foo%/%tile2% x%foo%%cover%
            &end
```

```
            &if [exists %cdpath%/dcw/sasaus/%foo%/%tile2% -
directory] AND ~
              ^ [exists x%foo%%cover% -cover] AND ~
              [exists x%foo%%cover%pnt -cover] &then &do
                  kill x%foo%%cover%pnt all
                  vpfimport cover
%cdpath%/dcw/sasaus/%foo%/%tile2% x%foo%%cover%
            &end
            &if [exists %cdpath%/dcw/sasaus/%foo%/%tile2% -
directory] AND ~
              [exists x%foo%%cover% -cover] AND ~
              ^ [exists x%foo%%cover%pnt -cover] &then &do
                  kill x%foo%%cover% all
                  vpfimport cover
%cdpath%/dcw/sasaus/%foo%/%tile2% x%foo%%cover%
            &end
         &end

         &do foo &list rr rd po           /* coverages with
other features
           &if [exists %cdpath%/dcw/sasaus/%foo%/%tile2% -
directory] AND ~
             ^ [exists x%foo%%cover% -cover] &then &do
                 vpfimport cover
%cdpath%/dcw/sasaus/%foo%/%tile2% x%foo%%cover%
               &end
         &end
      &end
   &end

&s junk [close %tilelist%]

&ty
&ty                            Done importing coverages.

/* *************************************************
/* Appending coverages and dissolving common borders
/* *************************************************
/* Variables, files, and coverages used:
/*

&s total %total_t1% + %total_t2% + %total_t3% + %total_t4%

&ty Combining coverages and dissolving common borders.

&s junk [close -all]   /* safety close
```

```
/* **************************************Working on points
first
&do foo &list dn hy hs of pp ds lc ae /* coverages with
point features

  /* Most of this section is deleted as JTS doesn't deal
with points

  /* Finally, delete all the individual point coverages
  &s tilelist [open xtilelist.txt os -read]
  &do loop = 1 &to %total%
    &s tile [read %tilelist% rs]
    &s a [locase [substr %tile% 1 1]]
    &s b [locase [substr %tile% 3 1]]
    &s c [locase [substr %tile% 5 2]]
    &if [exists x%foo%%a%%b%%c%pnt -cover] &then &do
        kill x%foo%%a%%b%%c%pnt all
      &end
  &end
  &s junk [close %tilelist%]

&end

/* **************************************Working on polygons
second
&do foo &list dn hy lc po pp /* coverages with polygon
features

  /* first, check to see how many coverages need to be
combined
  &s tilelist [open xtilelist.txt os -read]
  &s temptot 0
  &do loop = 1 &to %total%
    &s tile [read %tilelist% rs]
    &s a [locase [substr %tile% 1 1]]
    &s b [locase [substr %tile% 3 1]]
    &s c [locase [substr %tile% 5 2]]
    &if [exists x%foo%%a%%b%%c% -poly] &then
      &s temptot %temptot% + 1
  &end
  &s junk [close %tilelist%]

  /* second, actually combine or copy the coverages as
necessary
  /* if there is only one valid coverage, then copy it
  &if %temptot% = 1 &then &do
        &s tilelist [open xtilelist.txt os -read]
        &do loop = 1 &to %total%
          &s tile [read %tilelist% rs]
          &s a [locase [substr %tile% 1 1]]
          &s b [locase [substr %tile% 3 1]]
          &s c [locase [substr %tile% 5 2]]
          &if [exists x%foo%%a%%b%%c% -poly] &then &do
            cp x%foo%%a%%b%%c% x%foo%py
            &end
        &end
        &s junk [close %tilelist%]
    &end

  /* if there are more valid coverages, then use APPEND
  &else &if %temptot% gt 1 &then &do
        &s tilelist [open xtilelist.txt os -read]
        append x%foo%py poly
        &do loop = 1 &to %total%
          &s tile [read %tilelist% rs]
          &s a [locase [substr %tile% 1 1]]
          &s b [locase [substr %tile% 3 1]]
          &s c [locase [substr %tile% 5 2]]
          &if [exists x%foo%%a%%b%%c% -poly] &then
x%foo%%a%%b%%c%
        &end
        end      /* signifying the end of entering coverages
to be appended
        &s junk [close %tilelist%]
      &end

  &if [exists x%foo%py -cover] &then &do
      tolerance x%foo%py nodesnap .01
      clean x%foo%py  /* Polygon topology isn't maintained
through APPEND
    &end
&end

/* **************************************Working on lines
third
&do foo &list dn hy hs of po rr rd /* coverages with line
features

  /* check to be sure all coverages actually have lines and
not just
  /* line topology (to be sure the items will match)
  ae
  &s tilelist [open xtilelist.txt os -read]
  &s temptot 0
  &do loop = 1 &to %total%
    &s tile [read %tilelist% rs]
    &s a [locase [substr %tile% 1 1]]
    &s b [locase [substr %tile% 3 1]]
    &s c [locase [substr %tile% 5 2]]
    &if [exists x%foo%%a%%b%%c% -line] &then &do
        ec x%foo%%a%%b%%c%
        ef arc
        sel all
        &if [show number selected] = 0 &then
          kill x%foo%%a%%b%%c% yes
      &end
  &end
  &s junk [close %tilelist%]
  quit

  /* check to see how many coverages need to be combined
  &s tilelist [open xtilelist.txt os -read]
  &s temptot 0
  &do loop = 1 &to %total%
    &s tile [read %tilelist% rs]
    &s a [locase [substr %tile% 1 1]]
    &s b [locase [substr %tile% 3 1]]
    &s c [locase [substr %tile% 5 2]]
    &if [exists x%foo%%a%%b%%c% -line] &then
      &s temptot %temptot% + 1
  &end
  &s junk [close %tilelist%]

  /* combine or copy the coverages as necessary
  /* if there is only one valid coverage, then copy it
  &if %temptot% = 1 &then &do
        &s tilelist [open xtilelist.txt os -read]
        &do loop = 1 &to %total%
          &s tile [read %tilelist% rs]
          &s a [locase [substr %tile% 1 1]]
          &s b [locase [substr %tile% 3 1]]
          &s c [locase [substr %tile% 5 2]]
          &if [exists x%foo%%a%%b%%c% -line] &then &do
            cp x%foo%%a%%b%%c% x%foo%ln
            &end
        &end
        &s junk [close %tilelist%]
    &end
  /* if there are more valid coverages, then use APPEND
  &else &if %temptot% gt 1 &then &do
        &s tilelist [open xtilelist.txt os -read]
        append x%foo%ln line
        &do loop = 1 &to %total%
          &s tile [read %tilelist% rs]
```

```
          &s a [locase [substr %tile% 1 1]]
          &s b [locase [substr %tile% 3 1]]
          &s c [locase [substr %tile% 5 2]]
          &if [exists x%foo%%a%%b%%c% -line] &then
x%foo%%a%%b%%c%
        &end
        end      /* signifying the end of entering coverages
to be appended
        &s junk [close %tilelist%]
      &end

&end

&do foo &list ae dn hy hs of pp ds rr rd po lc
  /* Finally, delete all the individual line and poly
coverages
  &s tilelist [open xtilelist.txt os -read]
  &do loop = 1 &to %total%
    &s tile [read %tilelist% rs]
    &s a [locase [substr %tile% 1 1]]
    &s b [locase [substr %tile% 3 1]]
    &s c [locase [substr %tile% 5 2]]
    &if [exists x%foo%%a%%b%%c% -cover] &then &do
        kill x%foo%%a%%b%%c% all
      &end
  &end
  &s junk [close %tilelist%]
&end


/* Dissolving polygon coverages to get rid of lines at tile
boundaries
dissolve xhypy xtemp hypytype
kill xhypy all
rename xtemp xhypy

dissolve xlcpy xtemp lcpytype
kill xlcpy all
rename xtemp xlcpy

dissolve xpopy xtemp popytype
kill xpopy all
rename xtemp xpopy

dissolve xdnpy xtemp dnpytype
kill xdnpy all
rename xtemp xdnpy
```

```
dissolve xpppy xtemp pppytype
kill xpppy all
rename xtemp xpppy

/* clipping each coverage with the clip coverage and
/* changing each coverage name to include the desinated
group name
build xclipcov

&do foo &list ae dn hy hs of pp ds rr rd po lc

  &if [exists x%foo%ln -line] &then &do
      clip x%foo%ln xclipcov xtemp line
      kill x%foo%ln all
      rename xtemp %name%%foo%ln
    &end

  &if [exists x%foo%py -poly] &then &do
      clip x%foo%py xclipcov xtemp
      kill x%foo%py all
      rename xtemp %name%%foo%py
    &end

&end

/* Deleting tile boundary tiles in the hyln coverage; this
seems to be
/* the only place where it's a problem.
display 0
ae
ec %name%hyln
ef arc
sel hylntype lt 0
&if [show number selected] ne 0 &then delete
save
quit

&if %view% ne 'n' &then &do
    display 9999
    ae

    ec %name%popy
    ef arc
    de arc
    &if [exists %name%lcpy -cover] &then bc %name%lcpy 3
    &if [exists %name%pppy -cover] &then bc %name%pppy 8
    &if [exists %name%dnln -cover] &then bc %name%dnln 4
    &if [exists %name%rrln -cover] &then bc %name%rrln 6
    &if [exists %name%rdln -cover] &then bc %name%rdln 2
    be arc
    be poly
    draw
    &s junk2 [response 'Hit return when finished']

    quit
  &end

/* ***********
&ty Cleaning up...
/* ***********

/* Deleting used coverages

&do cov &list xtiles xtilesutm xclipcov xcliproj xtemp
   &if [exists %cov% -cover] &then kill %cov% all
&end

/* Projecting all the coverages to UTM in anticipation of
/* creating JTS terrain.

&s xmin %ulx%
&s ymin %lry%
&s xmax %lrx%
&s ymax %uly%

&if %xmin% gt 0 &then
  &s zone [calc [truncate [calc %xmin% / 6]] + 31]
&else &do
    &s zonemin [calc 180 + %xmin%]
    &s zone [calc [truncate [calc %zonemin% / 6]] + 1]
  &end

&do foo &list ae dn hy hs of pp ds rr rd po lc

  &if [exists %name%%foo%ln -line] &then &do
      project cover %name%%foo%ln xtemp
         output
         projection utm
         zone %zone%
         units meters
         spheroid wgs84
         parameters
         end
      kill %name%%foo%ln all
      rename xtemp %name%%foo%ln
      build %name%%foo%ln line
    &end

  &if [exists %name%%foo%py -poly] &then &do
      project cover %name%%foo%py xtemp
         output
         projection utm
         zone %zone%
         units meters
         spheroid wgs84
         parameters
         end
      kill %name%%foo%py all
      rename xtemp %name%%foo%py
      build %name%%foo%py
    &end

&end

/* **************
/* Now moving on to the JTS coverage construction part
/* **************

/* First, creating a workspace for the jts coverages
&s tester .TRUE.
&s counter 0
&do &while %tester%
  &if [exists jts%name%%counter% -workspace] &then
    &s counter %counter% + 1
  &else &do
      cw jts%name%%counter%
      &s tester .FALSE.
    &end
&end

/* Second, creating the clip coverage and empty attributed
feature
/* coverages necessary for JTS terrain construction.
w jts%name%%counter%

/* createclip xclipcov %ulx% %lry% %lrx% %uly%

/* inserting the aml createclip.aml here, with modifications


/* ========================================================
/* CREATECLIP.AML
/*
```

```
/* ATOOL to create clip coverages from user specified
coordinates.

&s cover xclipcov

&if [exists %cover% -cover] &then
   kill %cover% all
&if [exists %cover%2 -cover] &then
   kill %cover%2 all

generate %cover%
   polys
   2,[calc [calc [calc %xmax% - %xmin%] / 2] + %xmin%] ~
     [calc [calc [calc %ymax% - %ymin%] / 2] + %ymin%]
   %xmin% %ymin%
   %xmax% %ymin%
   %xmax% %ymax%
   %xmin% %ymax%
   %xmin% %ymin%
   end
   end
quit

project cover %cover% %cover%2
   input
   projection geographic
   units dd
   datum wgs84
   spheroid wgs84
   parameters
   output
   projection utm
   zone %zone%
   units meters
   datum wgs84
   spheroid wgs84
   parameters
   end

kill %cover% all
rename %cover%2 %cover%
build %cover% poly
/* =========================================================

zcreate xclipcov
copy paved rail   /* since it's not created by zcreate.aml
```

```
/* Third, opening up each DCW coverage individually and
copying features
/* into the JTS coverages
display 0
ae
&if [exists ../%name%rrln -line] &then &do       /* railroad
    ec ../%name%rrln
    ef arc
    sel all
    &if [show number selected] ne 0 &then &do
        put rail
        y
      &end
  &end
&if [exists ../%name%rdln -line] &then &do         /* roads
    ec ../%name%rdln
    ef arc
    sel rdlntype = 1     /* bahn
    &if [show number selected] ne 0 &then &do
        put bahn
        y
      &end
    sel rdlntype = 2     /* paved
    &if [show number selected] ne 0 &then &do
        put paved
        y
      &end
    sel rdlntype = 3     /* tracks
    &if [show number selected] ne 0 &then &do
        put track
        y
      &end
    sel rdlntype = 8     /* connectors
    &if [show number selected] ne 0 &then &do
        put extra
      &end
  &end
&if [exists ../%name%dnln -line] &then &do         /* rivers
    ec ../%name%dnln
    ef arc
    sel dnlntype = 1     /* normal rivers
    asel dnlntype = 4    /* canals and acqueducts
    &if [show number selected] ne 0 &then &do
        put hydro
        y
      &end
  &end
&if [exists ../%name%dnpy -poly] &then &do         /* lakes
    ec ../%name%dnpy
    ef poly
    sel dnpytype = 1
    asel dnpytype = 2
    &if [show number selected] ne 0 &then &do
        put lakes
        y
      &end
  &end
&if [exists ../%name%popy -poly] &then &do         /* oceans
    ec ../%name%popy
    ef poly
    sel popytype = 2
    &if [show number selected] ne 0 &then &do
        put lakes
        y
      &end
  &end
&if [exists ../%name%pppy -poly] &then &do         /* urban
    ec ../%name%pppy
    ef poly
    sel pppytype = 0
    &if [show number selected] ne 0 &then &do
        put urban
        y
      &end
  &end
&if [exists ../%name%lcpy -poly] &then &do     /* land cover
    ec ../%name%lcpy
    ef poly
    sel lcpytype = 14
    asel lcpytype = 15     /* swamp
    &if [show number selected] ne 0 &then &do
        put swamp
        y
      &end
    sel lcpytype = 3     /* vineyard
    &if [show number selected] ne 0 &then &do
        put vineyard
        y
      &end
    sel lcpytype = 6     /* fish ponds
    &if [show number selected] ne 0 &then &do
        put lakes
        y
      &end
  &end
quit
```

```
rename xclipcov clip

display 9999
w ..

/* Done.
&echo &off
&type Time to create JTS terrain!  Don't forget to take the
roads
&type that are in EXTRA and put them into the other roads
coverages.
&type Also, clean the boundaries between inland lakes and
oceans.
&ty [date -VFULL]
&watch &off

&return
```